REMARKS

Claims 21 and 31 are amended and claims 51-53 are added. Claims 1-53 are pending in the application. Reconsideration of the application is requested in view of the amendments and the remarks to follow.

The amendments to claims 21 and 31 address minor informalities noted during review, however, these amendments are not intended to alter the scope of the claims.

Applicant notes that the anticipation rejections and the rejections based on formal matters have not been repeated in the present Office Action. Applicant thus notes that these rejections have been overcome.

Allowed Subject Matter:

Claims 20, 21 and 31 are stated (Summary; page 7, item 8) to be allowed.

New Claims:

New claims 51-53 are supported at least by text appearing at p. 4, line 19 through p. 17, line 20 of the application as originally filed. No new matter is added by new claims 51-53. New claim 51 is similar to claim 20 but differs in scope. New claim 52 is similar to claim 21 but differs in scope. New claim 53 is similar to claim 31 but differs in scope. New claims 51-53 distinguish over the art of record and are allowable.

Art Rejections:

I.   Claims 1-6, 8-13, 15, 32, 33, 35-37, 39, 41-46, 48 and 50 are stated (p. 2) to stand rejected under 35 U.S.C. §103(a) as being unpatentable over Inside Macintosh (QuickDraw GX Environment and Utilities, Chs. 2 and 3), hereinafter "IM", in view of Gien et al., **"Micro-Kernel Based Operating Systems**; Moving Unix Onto Modern System Architecture" (emphasis added; ©1991 by Chorus Systemes, Saint-Quentin-en-Yvelines, France; Proceedings of the UniForum 1992 Conference, San Francisco, USA, January 22-24, 1992).

II.   Claims 7, 14, 16-19, 22, 47 and 49 are stated (p. 5) to stand rejected under 35 U.S.C. §103(a) as being unpatentable over IM in view of Gien and further in view of Lindholm et al., U.S. Patent No. 5,765,157 (hereinafter "Lindholm").

III.   Claims 23-28 and 30 are stated (p. 6) to stand rejected under 35 U.S.C. §103(a) as being unpatentable over IM in view of Gien and (apparently) further in view of Culbert et al., U.S. Patent No. 5,696,926 (hereinafter "Culbert").

IV.   Claims 34, 38 and 40 are stated (p. 6) to stand rejected under 35 U.S.C. §103(a) as being unpatentable over IM in view of Gien and (apparently) further in view of Berstis et al., U.S. Patent No. 5,909,215 (hereinafter "Berstis").   Claim 29 is stated (p. 7) to stand rejected under 35 U.S.C. §103(a) as being unpatentable over IM in view of Gien and Culbert and further in view of Lindholm.

In traversing the rejections, it is helpful to first review the descriptions provided within the references.

Traverse of rejection I under 35 U.S.C. §103:

With respect to all of rejections (I) through (IV), IM is a textbook describing software graphics tools developed to operate on Apple Computer Macintosh Operating Systems for individual personal computers.

More particularly, IM describes an application program, QuickDraw® software GX, that can be used in the Macintosh Operating System environment (see, e.g., Ch. 1, p. 1-3). Most QuickDraw® GX software functions are designed for implementation on any platform ("The Macintosh Interface", p. 1-3) but some portions are specific to the Macintosh environment or operating system (Id.). As noted on p. 2-3 ("About QuickDraw GX Memory Management"), "QuickDraw GX works with the Macintosh Memory Manager to manage the memory used by our application for creating and manipulating objects." As noted on p. 2-4 ("Graphic Clients and Graphics Client Heaps"), "QuickDraw GX never deallocates graphics client heaps while they are in use." (p. 2-4, 7TH full ¶, emphasis added) .

IM discloses errors, warnings and notices that can be posted by QuickDraw® GX software functions (p. 3-3, first sentence). IM discloses descriptions of how a user can obtain such errors, warnings and notices, change the QuickDraw® GX software errors, warnings and notices, ignore the QuickDraw® GX software errors, warnings and notices, and install application-defined error, warning and notice handlers (bulleted list, middle of page 3-3).

The Office Action states (p. 2) that "As to claim 1, IM teaches a method of controlling memory usage in a computer system having limited physical memory, wherein one or more application programs (application) execute in conjunction with an operating environment (**mac OS** and QuickDraw GX on which 'your

application' runs) ...." Such is in error on multiple grounds. Applicant notes with particularity that this statement distinguishes between the Macintosh® operating system ("OS") and QuickDraw® GX software. The assertions contained within the Office Action thus are internally inconsistent.

QuickDraw® GX programs facilitate <u>development</u> of applications (see, e.g., page 1-3, 1ST ¶ following "ABOUT QUICKDRAW GX AND THE MACINTOSH ENVIRONMENT", stating that "QuickDraw GX provides a number of useful functions that assist <u>your application development</u> on the Macintosh computer."). Not one of the cited pages describes QuickDraw GX software as comprising an operating system or as being software on which an application runs. Clarification is requested.

The Office Action then states "the method comprising:

setting a plurality of memory thresholds (thresholds for warning / graphics_client_memory_too_small, non-fatal errors / could_not_dispose_backing_store, fatal errors / out_of_memory) and

the operating environment wielding, at increasingly critical memory thresholds (from warning to non-fatal errors to fatal errors), correspondingly increasing control over said one or more applications programs to reduce memory usage (from continue execution to continue execution internally to terminate execution immediately). See pages 3-3; 3-7; 3-11; 3-41; 3-42; 3-45."

All programs terminate when a fatal error is detected. In this case, the QuickDraw® GX application is simply providing status indications with respect to <u>itself</u>. It is not an "operating environment", as stated in the Office Action, or an "operating system", as recited in claim 1. The characterization in the Office

Action is different from, and not the same as, "increasing control over said one or more application programs to reduce memory usage" by an operating system, as recited in claim 1. The nature of memory self-management employed by QuickDraw® GX software is described beginning at p. 2-10.

In contrast, Gien describes "Micro-kernel Based Operating Systems: Moving UNIX onto Modern System Architectures" (Title). Gien states (1ST ¶) paragraph" that "High-volume, low-end systems are driving the pricing and gather more of the attention of much of the industry, but the high-end systems are driving the technology innovations since the low-end systems are built from standard components. Those innovations in the high-end work their way down to the low end, just as multiprocessing, graphics, etc. have moved down to the PCs. Having a successful strategy in the high end is important for ensuring a continuing technology flow into the medium and low end systems."

Gien states (2ND ¶) that "To build these systems, system builders need tools and services that will let them more easily integrate complex hardware architectures into high performance, reliable, distributed environments. To master these more complicated environments and get these more complicated machines to market faster, system builders need an operating system development environment that is as powerful as the development environment provided to applications by the UNIX system."

The next (3RD) paragraph states that "The emphasis for such a development environment should be on two critical areas. First is enabling systems services which let the OS engineer focus on the specific high valued [sic] added features of the system and on the most efficient and flexible implementation of those features

and **not on how to shoe-horn those features into the existing code base.** System builders should be able to add value in several ways, such as adding TP, DB and performance monitors to the operating system, supporting multicomputer architectures, improving reliability, high availability and fault tolerance, or extending levels of security." (emphasis added).

Gien thus teaches away from the notions offered in the Office Action, viz., to shoe-horn the features for which Gien is cited by the Office Action into the existing code base for Macintosh computers. It is improper to combine the teachings of references where the references teach away from the proposed combination. This is explained below in more detail with reference to MPEP §2145 (X)(D)(2), entitled "References Cannot Be Combined Where Reference Teaches Away from Their Combination".

This MPEP subsection states that: "It is improper to combine references where the references teach away from their combination. *In re Grasselli*, 713 F.2d 731, 743, 218 USPQ 769, 779 (Fed. Cir. 1983)". As a result, all of the unpatentability rejections are prima facie defective and should be withdrawn, and Applicants' claims should be allowed.

Additionally, taking bits and pieces of the teachings of IM and/or Gien, where there is no teaching or guidance within either reference to select some portions but not others of their teachings, comprises an impermissible "obvious to try" standard of unpatentability guided solely by hindsight reconstruction using Applicants' own disclosure as a template.

Such a standard of unpatentability is improper, as is discussed below in more detail with reference to MPEP §2145(X)(B), entitled "Obvious To Try

Rationale". This MPEP section states that "The admonition that 'obvious to try' is not the standard under §103 has been directed mainly at two kinds of error. In some cases, what would have been 'obvious to try' would have been to vary all parameters or try each of numerous possible choices until one possibly arrived at a successful result, where the prior art gave either no indication of which parameters were critical or no direction as to which of many possible choices is likely to be successful. In others, what was 'obvious to try' was to explore a new technology or general approach that seemed to be a promising field of experimentation, where the prior art gave only general guidance as to the particular form of the claimed invention or how to achieve it." *In re O'Farrell*, 853 F.2d 894, 903, 7 USPQ2d 1673, 1681 (Fed. Cir. 1988) (citations omitted)".

In this instance, no guidance in selecting some but not others of the elements from the teachings of IM or Gien, with or without the other cited references is identified in the references. Similarly, no direction as to which of many possible choices is likely to be successful has been identified. As a result, the unpatentability rejections are prima facie defective and should be withdrawn, and Applicants' claims should be allowed.

As there is no basis for the Examiner's contentions within the cited references, the only possible motivation for these contentions is hindsight reconstruction wherein the Examiner is utilizing Applicants' own disclosure to construct a reason for combining and/or modifying the teachings of the cited references. The Examiner is reminded that hindsight reconstruction is not an appropriate basis for a §103 rejection. (See, e.g., *Interconnect Planning Corp. v. Feil*, 227 USPQ 543, 551 (Fed. Cir. 1985); *In re Mills*, 16 USPQ2d 1430 (Fed.

Cir. 1990) (explaining that hindsight reconstruction is an improper basis for rejection of a claim).) As a result, the unpatentability rejections are prima facie defective and should be withdrawn, and Applicants' claims should be allowed.

Furthermore, Gien is explicitly directed to harnessing the power of multiprocessor computers in a UNIX® operating system environment. The teachings of Gien are rendered unsuitable for their intended purpose in making such an adaptation. It is improper to modify the teachings of a reference in a fashion that renders the teachings of the reference unsuitable for their intended purpose, as is explained below in more detail with reference to MPEP §2143.01.

In a subsection entitled "THE PROPOSED MODIFICATION CANNOT RENDER THE PRIOR ART UNSATISFACTORY FOR ITS INTENDED PURPOSE", this MPEP portion states that "If proposed modification would render the prior art invention being modified unsatisfactory for its intended purpose, then there is no suggestion or motivation to make the proposed modification. *In re Gordon*, 733 F.2d 900, 221 USPQ 1125 (Fed. Cir. 1984) ...." Because combining the teachings of IM with the teachings of Gien to try to arrive at the subject matter of any of Applicants' claims defeats the intended purposes of Gien, it is improper to combine the teachings of these references in an attempt to find unpatentability.

Moreover, in contrast to the cited references, claim 1 recites " A method of controlling memory usage in a computer system having limited physical memory, wherein one or more application programs execute in conjunction with an operating system, the method comprising: setting a plurality of memory thresholds; and the operating system wielding, at increasingly critical memory

thresholds, <u>correspondingly increasing control over said one or more application programs to reduce memory usage</u>", which is not taught or disclosed by IM.

QuickDraw® GX software is not an operating system. An operating system is a software product that defines basic operations of a computer and that provides an environment in which applications programs, such as graphics manipulation software, may be executed.

In other words, equating QuickDraw® GX software, as described in IM, to an operating system, as recited in claim 1, gives the term "operating system" a meaning repugnant to the ordinary meaning of the term. Such is improper, as is explained in more detail below with reference to MPEP §608.01(o), entitled "Basis for Claim Terminology in Description" (see also MPEP §2111.01).

MPEP §608.01 states that "The meaning of every term used in any of the claims should be apparent from the descriptive portion of the specification with clear disclosure as to its import; and in mechanical cases, it should be identified in the descriptive portion of the specification by reference to the drawing, designating the part or parts therein to which the term applies. A term used in the claims may be given a special meaning in the description. No term may be given a meaning repugnant to the usual meaning of the term."

Applicants' specification describes examples of operating systems (see p. 5, line 24 et seq.). For example, "An operating system 44, executed by processor 40, is resident in and utilizes memory 42. H/PC 20 preferably runs the Windows® CE operating system from Microsoft Corporation. This operating system is a derivative of Windows® brand operating systems, such as Windows® 95, that is especially designed for handheld computing devices." Applicants' specification

also teaches, for example, that "The Windows® CE operating system is a multitasking operating system that allows <u>simultaneous execution of multiple applications 45</u>. The operating system employs a graphical user interface windowing environment that presents applications and documents in specially delineated areas of the display screen called "windows". Each window can act independently, including its own menu, toolbar, pointers, and other controls, as if it were a virtual display device. It is noted, however, that the handheld computing device may be implemented with other types of operating systems." (p. 6, line 7 et seq.).

Put another way, one definition of an operating system is found at p. 279 of Microsoft Press Computer Dictionary, 2ND Ed., published by Microsoft Press, A Division Of Microsoft Corporation, One Redmond Way, Redmond, Washington (copyright 1994). This definition is:

> **operating system** Abbreviated OS; sometimes called the executive. The software responsible for controlling the allocation and usage of hardware resources such as memory, central processing unit (CPU) time, disk space, and peripheral devices. The operating system is the foundation on which applications, such as word-processing and spreadsheet programs, are built. Popular operation systems include MS-DOS, the <u>Macintosh OS</u>, OS/2, Windows, Windows NT and UNIX.

In contrast, the same dictionary defines an application program at p. 23 as:

> **application** A computer program designed to help people perform a certain type of work. <u>An application thus differs from an operating system</u> (which runs a computer), a utility (which performs maintenance or general-purpose chores), and a language (with which computer programs are created). Depending on the work for which it was designed, <u>an application can manipulate</u> text, numbers, <u>graphics</u>, or a combination of these elements. Some application packages offer considerable computing power by focusing on a

single task, such as word processing; others, called integrated software, offer somewhat less power but include several applications, such as a word processor, a spreadsheet, and a database program.

As such, operating systems and application programs are not equivalent. Equating the QuickDraw® GX software described in IM to the operating system recited in claim 1 thus gives both of the terms "operating system" and "applications program" meanings repugnant to the ordinary meaning of these terms <u>and</u> is inconsistent with the descriptions of these terms as provided in Applicants' specification.

From a slightly different perspective, it may be seen that these definitions are mutually exclusive; things which are operating systems are not application software, and vice versa. Thus, either QuickDraw® GX software comprises an "operating system" as recited in claim 1, or QuickDraw® GX software comprises an application software package, but QuickDraw® GX software cannot be both an operating system and an application software package.

If one assumes, arguendo, that QuickDraw® GX software comprises an operating system, then the teachings regarding fatal and non-fatal errors and error messages noted in the Office Action (p. 2) relate to shutting down of an operating system by operation of the same operating system, and not to increasing control over an application software package, as recited in claim 1. On the other hand, if one assumes, arguendo, that QuickDraw® GX software comprises an applications program, then the <u>same</u> teachings regarding fatal and non-fatal errors and error messages noted in the Office Action relate to shutting down of an application program by operation of the same application software, which <u>again</u> is not what is recited in claim 1, and which does not substitute for "<u>the operating system</u>

<u>wielding</u>, at increasingly critical memory thresholds, <u>correspondingly increasing</u> <u>control over said one or more application programs to reduce memory usage</u>" as recited in claim 1.

Again, the assertions contained within the Office Action thus are internally inconsistent. Accordingly, the rejection of claim 1 (and of all of Applicants' claims) is defective and should be withdrawn, and claim 1 should be allowed.

Claim 10 recites "A computer-readable storage medium having instructions for controlling memory usage in a computer system having limited physical memory, wherein one or more application programs execute in conjunction with an operating system, the instructions being executable by the computer system to perform acts comprising: at a first memory usage threshold, requesting at least one of the application programs to close itself; and at a second memory usage threshold that is more critical than the first memory usage threshold, terminating at least one of the application programs without allowing its further execution", which is not taught, disclosed, suggested or motivated by IM or Gien, alone or in any proper combination.

In the Response dated August 14, 2003 (p. 29), Applicant notes that the Office Action refers to the discussion of claims 1, 3 and 4 as formulating a basis for rejection of claim 10, which oblique argument is repeated in the present Office Action (p. 4). As noted above and in the prior Response, IM fails to provide the subject matter recited in claim 1 with or without the teachings of any other references. The rejection of claims 3 and 4 (p. 4) indicates that "IM shuts down an application when the application poses severe enough memory problem. Page 3-

41." Applicant finds no such teaching on page 3-41. Clarification is **again** requested.

Page 3-41 includes Figure 3-5, entitled "Enabling and disabling an error handler". Error handlers are described on the preceding page, for example, as user-developed and installed applications subprograms that respond to warnings and notices from QuickDraw® GX software.

The text on page 3-41 states that:

> The handler should respond to the problems that occur during typical application scenarios. A friendly application should let the user know when it is taking action in response to errors and warnings that have occurred. For example, if an application runs out of memory and that it is responding in a particular manner to alleviate the problem. If it cannot solve the problem, it may need to notify the user that it needs to abort processing. Such an application would need to install an error handler that looks for out_of_memory errors.

> In general, in the non-debugging version of your application, the handler might be relatively simple. If the handler doesn't have a response to an error or warning, it should just return and continue execution.

> In contrast, the debugging version of the handler may be relatively complex to accommodate special error, warning, and notice conditions. In general, you should stop and print the errors, warnings, and notices whenever a problem occurs.

> An application can have more than one error handler. A simple application might have just one error handler to handle specific problems. However, a more complicated application may have multiple error handlers. For example, an application might have one error handler that takes care of memory problems and another error handler for other types of errors. The special error handler may be installed only when a particular type of processing is to occur, like animation or QuickTime movies.

This passage explicitly discusses error handlers that are a part of an application, and makes no mention or suggestion whatsoever of "requesting at least one of the application programs to close itself" at a first memory threshold, as affirmatively recited in claim 10. In fact, it makes no logical sense for "an application program" to be "requesting at least one of the application programs to close itself" - how does the Examiner propose to implement such? How would an application program assume such control of "at least one of the application programs"? How, indeed, does the Examiner propose that an application program even have information relative to "one or more application programs" sufficient to effectuate such?

This passage also fails to suggest or motivate "terminating at least one of the application programs without allowing its further execution" at a second memory threshold, as affirmatively recited in claim 10. The concerns noted above are also relevant to this clause of claim 10.

Further, notifying a user that a program may need to be shut down is not an equivalent to termination of a program without allowing further execution, as recited in claim 10. For at least these reasons, the rejection of claim 10 is defective and should be withdrawn, and claim 10 should be allowed.

Claim 32 recites "A method of controlling memory usage in a computer system having limited physical memory, wherein one or more application programs execute in conjunction with an operating system, the method comprising: monitoring memory usage; and when memory usage is high, sending a message from the operating system to at least one of the application programs

requesting the application program to reduce its current use of memory", which is not taught or disclosed by IM or Gien, alone or in any proper combination.

As noted above, IM does not teach an operating system, as affirmatively recited in claim 32. IM also does not teach or disclose sending a message from the operating system to at least one of the application programs requesting the application program to reduce its current use of memory, as recited in claim 32.

Claim 37 recites "A computer-readable storage medium having instructions for controlling memory usage in a computer system having limited physical memory, wherein one or more application programs execute in conjunction with an operating system, the instructions being executable by the computer system to perform acts comprising: monitoring memory usage; and at a defined memory usage threshold, sending a message from the operating system to at least one of the application programs requesting the application program to reduce its current use of memory", which is not taught or disclosed by IM or Gien, alone or in any proper combination.

As noted above, IM does not teach instructions executable by a processor to monitor memory usage or send a message <u>from the operating system</u> to <u>at least one of the application programs</u> requesting that the application program reduce current memory use, as recited in claim 37.

In further contrast, claim 41 recites "A computer-readable storage medium having computer-executable instructions embodied therein, that, when executed by a processor, cause the processor to execute a process for controlling memory usage in a computer system having limited physical memory, wherein one or more application programs execute in conjunction with an operating system, the

instructions being configured to cause the processor to: <u>set a plurality of memory</u> <u>thresholds</u>; and increase control over the one or more application programs <u>by the</u> <u>operating system</u> to reduce memory usage at increasingly critical memory thresholds, the instructions configured to cause the processor to increase control including instructions to cause the processor to request at least one of the application programs to limit its use of memory at a first memory threshold", which is not taught or disclosed by IM or Gien, alone or in any proper combination.

The references provide no teachings whatsoever regarding anything to cause an operating system to set memory thresholds, as recited in claim 41. The references also are silent with respect to control being wielded by an operating system responsive to memory thresholds, as recited in claim 41.

Additionally, claim 50 recites "A computer-readable storage medium having computer-executable instructions for performing a process for controlling memory usage in a computer system having limited physical memory, wherein one or more application programs execute in conjunction with an operating system, the process comprising: monitoring memory usage; and when memory usage is high, <u>sending a message from the operating system to at least one of the</u> <u>application programs</u> including a particular application program that was least recently active, requesting the application program to reduce its current use of memory", which is not taught or disclosed by IM or Gien, alone or in any proper combination.

The references are silent with respect to "sending a message from the operating system to at least one of the application programs", as recited in claim

50. The references are also silent with respect to the message "requesting the application program to reduce its current use of memory", as recited in claim 50. Additionally, because the references fail to provide the affirmatively-recited aspects of the subject matter of the claims, the Office Action fails to establish a prima facie case of unpatentability (see discussion of MPEP §2143, infra). For at least these reasons, the rejection of claims 1, 10 32, 37, 41 and 50 is prima facie defective and should be withdrawn, and claims 1, 10, 32, 37, 41 and 50 should be allowed.

Traverse of rejection II under 35 U.S.C. §103:

Lindholm describes a "Computer system and method for executing threads of execution with reduced run-time memory space requirements" (Title). Lindholm is directed to "A computer system and associated method for executing a plurality of threads of execution with reduced memory space requirements. The computer system comprises a memory, an execution controller, and a data compressor. The execution controller controls execution of the threads such that the threads are executable and unexecutable at different times. The execution controller also stores uncompressed into available space in the run-time memory execution data of the threads when the execution data is generated. The data compressor compresses the uncompressed execution data of compressible ones of the threads that are unexecutable. As a result, space is made available in the run-time memory. The data compressor also decompresses in available space in the run-time memory the compressed execution data of decompressible ones of the threads so that the decompressible ones of the threads may be executed after becoming executable." (Abstract).

Claim 7 recites "A method as recited in claim 1, further comprising: at one or more of the memory thresholds, reclaiming unused stack memory", which is not taught, disclosed, suggested or motivated by the cited references, alone or in any proper combination.

Claim 17 recites "A method of controlling memory usage in a computer system having limited physical memory, wherein one or more application programs execute in conjunction with an operating system, the method comprising: at a first memory usage threshold, requesting at least one of the

application programs to limit its use of memory; at a second memory usage threshold that is more critical than the first memory usage threshold, requesting at least one of the application programs to close itself; at a third memory usage threshold that is more critical than the first and second memory usage thresholds, terminating at least one of the application programs without allowing its further execution; and reclaiming unused stack memory and discarding read-only memory before requesting at least one of the application programs to close itself and before terminating at least one of the application programs", which is not taught, disclosed, suggested or motivated by the cited references, alone or in any proper combination.

IM does not provide any teaching, disclosure, suggestion or motivation for requesting an application program to limit memory usage at a first memory usage threshold, as affirmatively recited in claim 17. IM also does not provide any teaching, disclosure, suggestion or motivation for requesting an application to close itself at a second memory usage threshold more critical than the first memory usage threshold. IM also does not provide any teaching, disclosure, suggestion or motivation for terminating at least one of the application programs without allowing its further execution at a third memory usage threshold more critical than the first or second memory usage thresholds, as recited in claim 17.

Claim 49 recites "A computer-readable storage medium having computer-executable instructions for performing a process for controlling memory usage in a computer system having limited physical memory, wherein one or more application programs execute in conjunction with an operating system, the process comprising: at a first memory usage threshold, requesting at least one of the

application programs to limit its use of memory; at a second memory usage threshold that is more critical than the first memory usage threshold, requesting at least one of the application programs to close itself; at a third memory usage threshold that is more critical than the first and second memory usage thresholds, terminating at least one of the application programs without allowing its further execution, and, before performing the terminating, prompting the user to select a currently executing application program to be terminated; and reclaiming unused stack memory and discarding read-only memory before requesting at least one of the application programs to close itself and before terminating at least one of the application programs", which is not taught, disclosed, suggested or motivated by the cited references, alone or in any proper combination.

The Office Action states (p. 5) that "Lindholm teaches memory management, including at a memory threshold, reclaiming unused stack memory (deallocate memory area of ANC stack that is no longer needed). Col. 8, lines 6-22." The cited passage is reproduced below:

> Similarly, while the thread 200 is executable (step 310 of FIG. 3), each time that it no longer requires a portion of its memory space (step 314 of FIG. 3), the memory space of the thread is decreased by deallocating the portion that it is not needed (step 316 of FIG. 3). Specifically, the size of the ANC stack is decreased whenever less of it is required by the thread during its execution. For instance, if memory is being allocated in 4K byte blocks, when the portion of the ANC stack used by the thread decreases by enough to allow the deallocation of a 4K byte block (i.e., the pointer to the top of the ANC stack points to a location in the preceding 4K block of the ANC stack), then the size of the ANC is decreased by one 4K byte block. This is done by deallocating the ANC memory are a 208 of the ANC stack that is no longer needed. In deallocating this ANC memory area, the execution controller updates the thread storage status table 210 so that the ANC memory area is removed from it and is no longer identified as being part of the ANC stack.

This passage is notably void of any mention of threshold or memory threshold. In fact, the entirety of Lindholm is void of the term "threshold" and thus Lindholm cannot possibly provide teachings relative to reclaiming unused stack memory at any memory threshold, as recited in claim 7.

The Office Action also states (p. 5) that "In doing so, the average run-time storage cost to support a program is reduced. (Lindholm, col. 2, lines 42-62)." The cited passage is included in the passage (lines 32-62) reproduced below:

> For such computer systems, low price is extremely important. In practice, one of the most significant costs in building such computer systems is the amount of run-time (i.e., run-time) memory that is required to run the software infrastructure and programs. This is to be distinguished from the static memory which is needed for storing the code of the software infrastructure and programs. It is very important to reduce the amount of run-time memory required by the kinds of computer systems just discussed since such a reduction produces a strong competitive advantage.
>
> Specifically, in multi threaded environments, a thread is typically implemented in part using one or more private areas (i.e., ranges) of run-time (i.e., run-time) memory to store the execution data needed to execute the thread. These private areas are typically in the form of stacks, heaps, or individual thread-local variables and represent the run-time storage cost of the thread. Thus, the run-time storage cost of a program is the sum of the run-time storage costs of all of its threads while the static storage cost of the program is the amount of memory used to store its code. Thus, it is desirable to reduce the average run-time storage cost of each thread in order to reduce the storage requirements of a computer system needed to support a program.

This passage provides no motivation for deallocation of stack memory responsive to memory thresholds. As noted above, Lindholm is void of any mention of any threshold or memory threshold. As such, the proposed combination fails to provide the subject matter recited in claim 7 and similarly fails to provide the subject matter of any of claims 14, 16-19, 22, 47 and 49. For

at least these reasons, the rejection of claims 7, 14, 16-19, 22, 47 and 49 is prima facie defective and should be withdrawn, and claims 7, 14, 16-19, 22, 47 and 49 should be allowed.

Traverse of rejection III under 35 U.S.C. §103:

Culbert describes a "Method and apparatus for transparently compressing data in a primary storage device" (Title). Culbert is directed to "A system for extending the data capacity of a primary storage device in a computer which utilizes transparent data compression. The system utilizes a compressed memory even within the primary storage device along with a compression/decompression algorithm to extend the memory space. The system also includes a water line or limit within the primary storage device to ensure that memory space is always available." (Abstract).

Claim 23 recites "A computer system comprising: a processor; an operating system that is executable by the processor and that utilizes the physical memory; <u>a</u> <u>virtual memory system that includes physical memory but does not include secondary storage</u>; one or more application programs that utilize the virtual memory system; wherein the operating system is configured to perform the following acts: <u>monitoring physical memory usage</u>; and <u>at increasingly critical physical memory usage thresholds, wielding increasing control over said one or more application programs to reduce physical memory usage</u>", which is not taught, disclosed, suggested or motivated by the cited references, alone or in any proper combination.

The Office Action states (p. 6) that Culbert teaches that an operating environment such as Mac operating system with QuickDraw® software functionality (col. 7, lines 18-31) is implemented on a computer system wherein a secondary storage is optional (col. 4, lines 56-67)." The description of QuickDraw® software provided by Culbert does not misidentify the QuickDraw®

application software as operating system software. To clarify what Culbert does

or does not state about such application software, the cited passage is reproduced

below:

> It will be noted there is a liberal use of graphic elements in the
> present invention. For example, the header bars 56a and 56b include
> lines and other graphical elements. Processes for drawing lines on a
> computer screen are well known to those skilled in the art. For
> example, graphics software such as QUICKDRAW from Apple
> Computer, Inc. of Cupertino, Calif. can be used to draw lines, simple
> geometrical shapes, etc. A description of the QUICKDRAW
> graphics software is found in the book Inside Macintosh, Volumes I,
> II, and III, by C. Rose et al., Addison-Wesley Publishing Company,
> Inc., July 1988. With such graphics software, a line can be drawn by
> simply specifying the coordinates of the beginning and the end of the
> line, and by specifying the width of the line.

This passage makes no characterization of the QuickDraw® application

software as comprising any "operating system" as alleged in the Office Action.

The passage cited in the Office Action (p. 6) in col. 4, lines 56-67 of Culbert is

reproduced below:

> Some type of mass storage 22 is generally considered desirable.
> Mass storage 22 can be coupled to I/O circuitry 18 by a bi-
> directional data bus 40. However, the mass storage 22 can be
> eliminated by providing a sufficient amount of RAM 16 to store user
> application programs and data. In that case, the RAM 16 can be
> provided with a backup battery to prevent the loss of data even when
> the pen-based computer system 10 is turned off. However, it is
> generally desirable to have some type of long term mass storage 22
> such as a commercially available miniature hard disk drive,
> nonvolatile memory such as flash memory, battery backed RAM, a
> PCMCIA card, or the like.

This passage merely states that RAM with battery backup may replace a

dedicated mass storage unit 22 such as a disk drive etc. to provide secondary

storage. It does not affirmatively recite "a virtual memory system that includes

physical memory but does not include secondary storage" (see p. 2, lines 4-9 of

the specification as filed for clarification), as recited in claim 23.

As noted above, IM fails to describe any operating system and instead describes an application. Culbert fails to cure the deficiencies of IM. None of IM, Gien or Culbert describe any operating system configured to monitor memory usage, as recited in claim 23. None of IM, Gien or Culbert describe any operating system configured to wield control over any application program to reduce physical memory usage, as recited in claim 23. As a result, it is inconceivable that combining the teachings of these references could provide the subject matter recited in claim 23.

Accordingly, Culbert does not provide the teachings for which it is cited. As a result, the rejection of claims 23-28 and 30 is prima facie defective and should be withdrawn, and claims 23-28 and 30 should be allowed.

Traverse of rejection IV under 35 U.S.C. §103:

Berstis describes a "method and apparatus to intercept and process error messages in a data processing system" (Title). Berstis states (Abstract) that "A process in a data processing system for handling messages received in a message queue in a message handling process for a graphical user interface. In response to receiving a message in the queue, a determination is made as to whether the error is a message. In response to an identification of an error message, that message is intercepted before processing by the message handling process. A determination is made as to whether a corrective action is required in response to the error message and as to whether the error message should be reformatted. If corrective action is required, that action is initiated by the process. Additionally, if the message should be reformatted, the process then reformats the message and returns the message to the message handling process for further processing."

Claim 40 recites "An application program that resides in a computer-readable memory for execution by a processor in conjunction with an operating system, the application program having a message loop that receives messages from an operating system, the application program being responsive to a particular message received through its message loop to reduce its current use of memory", which is not taught, disclosed, suggested or motivated by the cited references.

As noted above, IM fails to teach an application program receiving messages from an operating system. Berstis is directed to an error messaging system for use with computers using IBM-compatible operating systems (col. 5, lines 10-16).

Berstis is silent with respect to error, or any other, messages involving memory. There is no teaching or disclosure in Berstis, IM or Gien regarding any "application program being responsive to a particular message received through its message loop to reduce its current use of memory", as affirmatively recited in claim 40. As such, the proposed combination fails to provide the subject matter recited in claim 40. Additionally, the Office Action states (p. 7) that the motivation to combine the teachings of IM and Berstis comprises "providing the user with a useful error message indicating required corrective action(s) Berstis, col. 1, lines 51-60, col. 2, lines 6-21)". Claim 40 recites "the application program being responsive to a particular message received through its message loop to reduce its current use of memory" and is not concerned with "providing the user with a useful error message". As such, the argument provided in the Office Action is inapposite to the subject matter of claim 40. Clarification is requested.

The Office Action fails to establish a prima facie case of obviousness for <u>any</u> of the unpatentability rejections. Applicant notes that criteria for such are set forth in MPEP §2143, entitled "Basic Requirements of a Prima Facie Case of Obviousness" (see also MPEP §706.02(j)).

This MPEP section states that "To establish a prima facie case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings." (see also MPEP §2143.01, entitled "Suggestion or Motivation to Modify the References"). No motivation or guidance has been

identified <u>in the references</u> by the Office Action to modify the disclosure of the reference.

MPEP §2143 also states that "Second, there must be a reasonable expectation of success." (see also MPEP §2143.02, entitled "Reasonable Expectation of Success Is Required"). MPEP §2143 further states that "Finally, the prior art reference (or references when combined) must teach or suggest <u>all</u> the claim limitations." (see also MPEP §2143.03, entitled "All Claim Limitations Must Be Taught or Suggested"). As noted above, the references fail to teach or suggest all of the recitations of Applicants' independent claims, and particularly fail to provide any teaching or disclosure of numerous affirmatively-recited aspects of the subject matter of any of independent claims 10, 17, 23, 32, 37 or 40. As such, there can be no reasonable expectation of success.

This MPEP section further states that "<u>The teaching or suggestion</u> to make the claimed combination and the reasonable expectation of success <u>must both be found in the prior art</u>, not in applicants' disclosure. *In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991)."

This requirement is also described in MPEP §2143.01, entitled "Suggestion or Motivation To Modify the References." This MPEP portion includes a subsection stating that "THE PRIOR ART MUST SUGGEST THE DESIRABILITY OF THE CLAIMED INVENTION".

Inasmuch as the references fail to provide the subject matter of the independent claims, it is inconceivable that they could suggest the desirability of this subject matter. As a result, the rejection fails all prongs of the test set forth in the MPEP for a prima facie finding of unpatentability. For at least these reasons,

the unpatentability rejections should be withdrawn, and Applicants' claims should be allowed.

Moreover, no evidence has been provided as to why it would be obvious to combine or modify the teachings of these references. Evidence of a suggestion to combine or modify may flow from the prior art references themselves, from the knowledge of one skilled in the art, or from the nature of the problem to be solved. However, this range of sources does not diminish the requirement for _actual_ _evidence_. Further, the showing must be _clear and particular_. See _In re Dembiczak_, 175 F.3d 994, 998 (Fed. Cir. 1999).

Applicants' dependent claims distinguish for their own recited features and by virtue of dependence from allowable claims. Accordingly, the unpatentability rejections of claims 1-19, 22-30 and 32-50 are defective and should be withdrawn, and claims 1-19, 22-30 and 32-50 should be allowed.

Deficiencies in Examination:

The Examiner's response to argument is deficient in multiple regards. A first deficiency is that the response to argument clearly fails to respond to Applicants' arguments with respect to the rejections based on IM, or, in the alternative, is an admission that these rejections are defective.

Applicant notes the requirements of MPEP §707.07, entitled "Completeness and Clarity of Examiner's Action". This MPEP section cites 37 CFR §1.104, entitled "Nature of examination" which in turn states, in subsection (b), entitled "Completeness of examiner's action" that "The examiner's action will be complete as to all matters, except that in appropriate circumstances, such as misjoinder of invention, fundamental defects in the application, and the like, the action of the examiner may be limited to such matters before further action is made."

This MPEP section further states, under a heading labeled "Examiner Note" that "The Examiner must, however, address any arguments presented by the applicant which are still relevant to any references being applied." The Office Action clearly fails to comport with these requirements as set forth in the MPEP, at least because the Office Action both fails to address Applicants' arguments with respect to IM, with or without other references, and continues to reject claims over IM. Such is not consistent with promoting compact prosecution.

A second deficiency is that the combinations used in the unpatentability rejections fail to provide all of the features recited in any of Applicants' independent claims. The Examiner has ignored these features without providing any appropriate legal basis for doing so.

A third deficiency is the failure to respond to all arguments with respect to the references and particularly to fail to respond to the repeated requests for clarification. Merely repeating that "it would be obvious" to provide the features recited in the claims does not constitute a basis for rejection of the claims, particularly when the references fail to provide the features recited in the claims and the rejections fail to meet the standards for such rejections as set forth in the MPEP and as demonstrated by Applicant.

A fourth deficiency is to combine the teachings of disparate references absent any guidance in the references to support the combination when main intentions of each of the cited references are defeated by the combination and the references teach away from each other and from the claimed subject matter.

For at least these reasons, the Office Action fails to comport with appropriate standards for examination. The Examiner should either allow Applicants' claims or provide a meaningful basis for rejection and an appropriate response to Applicants' arguments.

Conclusion:

To recapitulate, Applicant traverses the rejections because (i) combining the teachings of the references fail to provide the claimed subject matter, (ii) the terminology of the references must be given meanings repugnant to the ordinary meanings of these terms as they are employed in the relevant arts to attempt to find unpatentability <u>and</u> is inconsistent with the descriptions of these terms as provided in Applicants' specification; (iii) the rejections contain multiple internal logical inconsistencies; (iv) the rejections fail to meet the standards set forth in the MPEP for a prima facie finding of unpatentability; (v) the references teach away from each other and from the claimed subject matter; (vi) the references are rendered unsuitable for their intended purposes in attempting to adapt their teachings to try to arrive at the claimed subject matter; (vii) the rejections employ an improper 'obvious to try' standard in attempting to find unpatentability; (viii) the rejections are based on an inapposite hindsight reconstruction; (ix) no <u>evidence</u> of motivation to combine or modify has been identified; (x) no motivation to combine/modify is found <u>in the references.</u>

Additionally, (xi) the rejections merely repeat arguments with respect to the teachings of the references that have been rebutted; (xii) the rebuttal and arguments relative to repeatedly applied reference materials has been improperly ignored <u>whilst repeatedly rejecting the claims and citing the same reference materials;</u> and (xiii) the prosecution history of the application is inconsistent with compact prosecution and with Applicants' <u>right</u> to seek intellectual property protection in furtherance of the constitutional purposes of the U.S. Patent Laws "to

promote the progress of science and useful arts." U.S. Constitution, Article 1, Section 8.

Claims 1-53 are in condition for allowance. Applicant respectfully requests reconsideration and issuance of the subject application. Should any matter in this case remain unresolved, the undersigned attorney respectfully requests a telephone conference with the Examiner to resolve any such outstanding matter.

Respectfully Submitted,

Date: _Mar. 30, 2004_   By: _____
                                    Frederick M. Fliegel
                                    Reg. No. 36,138
                                    (509) 324-9256 x239